

## **ГИПЕРТЕКСТОВОЕ ПРЕДСТАВЛЕНИЕ СЕМАНТИЧЕСКИХ СЕТЕЙ И ОНТОЛОГИЙ**

В связи с развитием информационных технологий большое значение приобретает накопление и систематизация используемых вычислительными системами данных и знаний, представленных в виде текстов. При такой систематизации важно учитывать не столько синтаксические особенности представленных текстами данных, сколько их семантические свойства, так как зачастую конечного пользователя интересует не форма представленной информации, а ее смысл. С этой целью разрабатываются и развиваются соответствующие средства — языки представления знаний и описания онтологий, которые позволяют

семантически структурировать представленные в вычислительных системах и глобальных вычислительных сетях данные. Существует множество языков, которые позволяют представлять знания или описывать онтологии. Одним из таких средств являются sc-языки (построенные на основе Semantic Code [2]), относящиеся к однородным семантическим сетям и обладающие следующим набором преимуществ перед другими языками: небольшим алфавитом, несложным синтаксисом, возможностью представления и интеграции различных видов знаний, включая декларативные, императивные, достоверные, нечеткие и многие другие [3]. Такие sc-языки ориентированы в первую очередь на представление знаний во внутренней памяти информационной системы, для ввода же текстов в память, хранящую тексты sc-языков, необходимо использовать способы представления и передачи знаний, которые являются удобными для пользователя. В частности, такими способами являются специальные языки Semantic Code graphic (SCg) и Semantic Code string (SCs) [2; 4]. Однако эти языки требуют некоторого времени на их изучение и потому не всегда удобны или привычны для пользователей глобальных сетей — Internet. Привычным и естественным для интернет-пользователей является гипертекстовое представление информации. Для обеспечения возможности семантической структуризации информации средствами sc-языков пользователям глобальных сетей необходимо предоставить соответствующие средства, которые были бы представимы в традиционном гипертекстовом виде с одной стороны, и были бы легко конвертируемы в конструкции sc-языков.

С этой целью предлагается гипертекстовый способ представления баз знаний в виде однородных семантических сетей. Гипертекстовый способ представления SCn (Semantic Code natural) — множество всевозможных sc.n-текстов. Каждый sc.n-текст состоит из sc.n-статей. Каждая sc.n-статья состоит из идентификатора, описываемого в этой sc.n-статье sc-элемента, и, возможно, одного или нескольких последующих sc.n-полей (рис. 1). При описании sc-элемента в sc.n-статье sc.n-поля описывают как, какими ролями и связками каких отношений связан описываемый sc-элемент с другими sc-элементами (элементами текстов sc-языков). Для описания SCn разработана модель гипертекстового представления исходных текстов баз знаний.

В модели гипертекстового представления исходных текстов баз знаний можно выделить две составляющие: структурную модель и модель семантики. Структурная модель гипертекстового представления исходных текстов баз знаний задается схемой, в которую входят следующие компоненты:

$U$  — множество адресных имен,

$T_{SCn}$  — множество sc.n-текстов ( $T_{SCn} \subseteq 2^{A_{SCn}}$ ),

$A_{SCn}$  — множество sc.n-статей,

$H_{SCn}$  — множество справочных sc.n-статей ( $H_{SCn} \subseteq A_{SCn}$ ),

$F_{SCn}$  — множество sc.n-полей,

$C_{SCn}$  — множество видов sc.n-полей ( $C_{SCn} = \{general\} \cup S_{SCn}$ ), где  $S_{SCn}$  — множество специальных видов sc.n-полей,

$FC_{SCn}$  — функция принадлежности sc.n-поля своему виду ( $FC_{SCn} = C_{SCn}^{F_{SCn}}$ ),  
 $E_{SCn}$  — множество идентификаторов понятий (sc-элементов),



Рис. 1. Фрагмент онтологии SCn-кода на языке SCg.

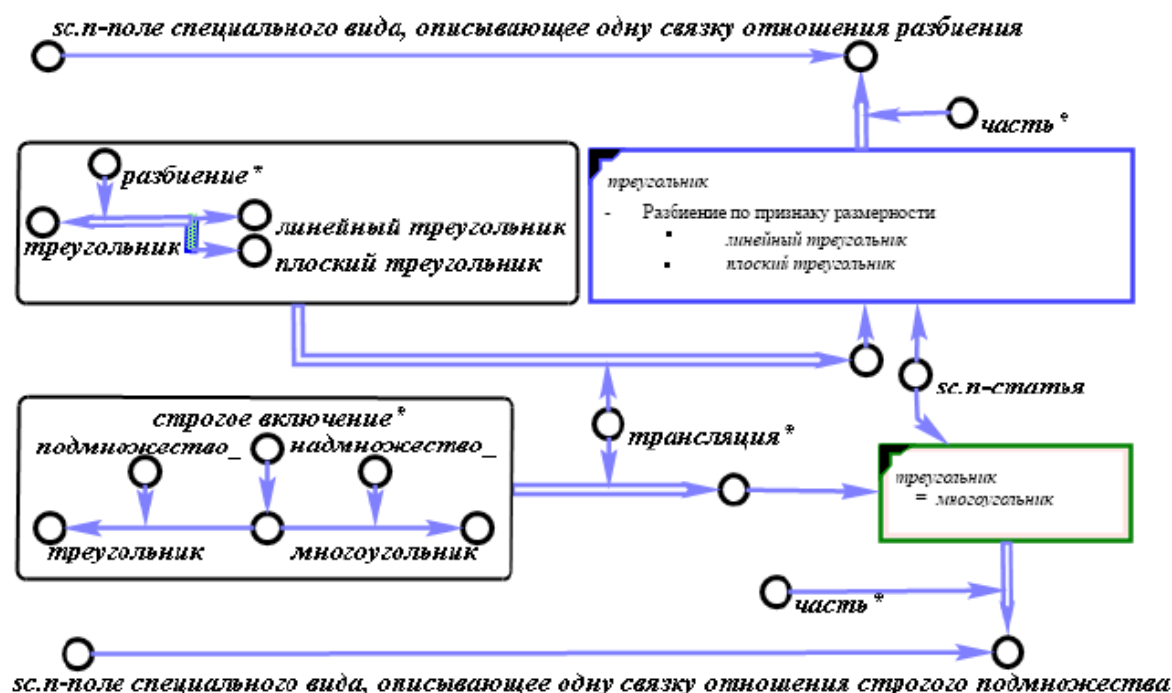


Рис. 2. Пример представления на SCn и эквивалентного представления на SCg.

$K_{SCn}$  — множество идентификаторов описываемых основными sc.n-статьями понятий ( $K_{SCn} \subseteq E_{SCn}$ ),  
 $R_{SCn}$  — множество идентификаторов sc-ссылок ( $R_{SCn} \subseteq E_{SCn}$ ),  
 $L_{SCn}$  — множество гиперссылок ( $L_{SCn} \subseteq E_{SCn} \times U$ ),  
 $I_{SCn}$  — отображение адресных имен внутренних гиперссылок на множество описываемых понятий ( $I_{SCn} \in K_{SCn}^U$ ),  
 $EO_{SCn}$  — функция отношений порядка идентификаторов понятий в sc.n-полях ( $EO_{SCn} \subseteq F_{SCn} \times (E_{SCn} \times E_{SCn})$ ),  
 $FO_{SCn}$  — функция отношений порядка sc.n-полей в sc.n-статьях ( $FO_{SCn} \subseteq A_{SCn} \times (F_{SCn} \times F_{SCn})$ ),  
 $AO_{SCn}$  — функция отношений порядка sc.n-статей в sc.n-текстах ( $AO_{SCn} \subseteq T_{SCn} \times (A_{SCn} \times A_{SCn})$ ),  
 $P_{SCn}$  — отношение участия идентификаторов понятий в sc.n-полях ( $P_{SCn} \in E_{SCn}^{F_{SCn}}$ ),  
 $D_{SCn}$  — отношение описания понятий sc.n-полями ( $D_{SCn} \in E_{SCn} \times F_{SCn}$ ),  
 $EP$  — множество точек входа в sc.n-тексты ( $EP \in K_{SCn}^U$ ),  
 $M_{SCn}$  — множество режимов отображения sc.n-текстов.

Кроме того, следует выделить множество  $B_{SCn}$  — множество блоков (частей), составляющих sc.n-статьи ( $B_{SCn} = K_{SCn} \cup F_{SCn}$ ). Обозначения вида  $2^M$ ,  $R^D$ ,  $R_+^D$  здесь и далее соответственно выражают множество всех подмножеств множества  $M$ , множество полных функций с областью определения  $D$  и  $R$  областью значений, множество полных и частичных функций с областью определения  $D$  и областью значений  $R$ .

На базе структурной модели гипертекстового представления знаний строится язык SCn, который имеет свой алфавит и синтаксис: в качестве символов алфавита используется подмножество символов из набора Unicode, синтаксис же имеет иерархическую структуру, в которой sc.n-поля, описывающие понятия, форматируются с помощью отступов в зависимости от положения идентификатора описываемого понятия. Положение идентификатора зависит и соответствует положению sc.n-поля, в котором он находится, при условии, что этот идентификатор не принадлежит множеству  $K_{SCn}$ , в последнем же случае он находится в самом крайнем положении. Для синтаксического анализа таким образом структурированных sc.n-текстов применима контекстно-зависимая грамматика, так как тексты имеют вид  $ax^nbx^mc$ , где  $b$  не содержит отступов  $x$ , а  $1 \leq m \leq n+1$ . Каждое sc.n-поле располагается в прямоугольной области, где начало каждой из строк sc.n-поля имеет одинаковый отступ от края. Проективная семантика [5] SCn на язык SC описывается моделью семантики гипертекстового представления исходных текстов баз знаний (рис. 2). Семантика SCn задаётся блоками sc.n-статей, в рамках контекстов. Каждый контекст является



соответствием между множеством идентификаторов контекстных величин *Identifiers* и их значений *Values*,  $X$  — множество всех контекстов.

$$X = \{Identifiers\} \times 2^{Identifiers \times Values} \times \{Values\} \quad (1)$$

Множество всех семейств контекстов соответственно можно обозначить как  $C = 2^X$ . Перевод с языка SCn на SC (в формате SCs) задаётся двумя функциями  $r$  и  $t$ .

$$\begin{aligned} r &: B_{SCn} \times C \rightarrow C \\ t &: (B_{SCn} \cup \{\emptyset\}) \times C \rightarrow (T_{SCs} \cup \{\emptyset\}) \end{aligned} \quad (2)$$

Первая из которых, начиная с начального множества контекстов  $c_0$ , позволяет построить последовательность множеств контекстов  $c_0, c_1, c_2, \dots, c_n$  для набора блоков  $b_1, b_2, \dots, b_n$ .

$$c_0 = \langle Identifiers, \emptyset, Values \rangle \quad (3)$$

Вторая функция  $t$  позволяет построить для каждой пары  $\langle c_k, b_{k+1} \rangle$  ( $b_{n+1} = \emptyset$ ) некоторый, возможно пустой, текст SCs из множества текстов  $T_{SCs}$ . Каждому блоку в sc.n-статье сопоставляется правило, которое анализирует каждый контекст из текущего множества контекстов и строит новые контексты, объединяя их в новое множество. Правила построения таких текстов описываются на языке, который позволяет представить выражения из множества *Expressions*, состоящие из следующих классов операторов.

Операторы класса преобразования (увеличения, уменьшения) числовых значений.

$$\begin{aligned} Increment &: Values \rightarrow Values_+ \\ Decrement &: Value \rightarrow Values_+ \end{aligned} \quad (4)$$

Операторы класса преобразования строковых и списковых значений.

$$\begin{aligned} Attach &: Values \times Values \times (Values \times Values) \rightarrow Values_+ \\ Detach &: Values \times (Values \times Values) \rightarrow (Values \times Values) \cup Values_+ \end{aligned} \quad (5)$$

Операторы класса поиска позиций и измерения списков и строк.

$$\begin{aligned} Locate &: Values \times Values \rightarrow (Values \times Values) \cup Values_+ \\ Length &: Values \rightarrow Values_+ \end{aligned} \quad (6)$$

Операторы класса получения по позиции элемента строки или списка.

$$At : Values \times Values \rightarrow Values_+ \quad (7)$$

Логические предикативные операторы.

$$\begin{aligned} Equality &: Values \times Values \rightarrow Values \\ Less &: Values \times Values \rightarrow Values_+ \end{aligned} \quad (8)$$

Логические операторы.

$$\begin{aligned} And &: Values \times Values \rightarrow Values_+ \\ Not &: Values \rightarrow Values_+ \end{aligned} \quad (9)$$

Условный оператор.

$$If : Values \times Expressions \times Expressions \rightarrow Expressions \quad (10)$$

Оператор вычисления значений идентифицированной величины.

$$Value : Identifiers \times X \rightarrow Values \quad (11)$$

Операторы инициализации и удаления идентифицированных величин.

$$\begin{aligned} Create &: Identifiers \times X \rightarrow X \\ Assign &: Identifiers \times Values \times X \rightarrow X \\ Erase &: Identifiers \times X \rightarrow X \\ Destroy &: Identifiers \times X \rightarrow X \end{aligned} \quad (12)$$

Идентификатор произвольного контекста из текущего множества контекстов.

$$CurrentOne \in C \quad (13)$$

Оператор объединения контекстов во множество ( $X \subseteq Values$ ).

$$Union : X \times X \rightarrow C \quad (14)$$

Оператор определения новых операторов при  $Expressions \subseteq Values$  семантически эквивалентен оператору *Assign*.

$$Define : Identifiers \times Expressions \times X \rightarrow X \quad (15)$$

На основе разработанных моделей с помощью платформы Mediawiki реализован набор шаблонов Mediawiki, которые в соответствии с правилами описания семантики sc.n-блоков одновременно с построением sc.n-текста (рис. 3) генерируют эквивалентный текст, который встраивается на HTML-страницу вместе с sc.n-текстом. Таким образом, обеспечивается простота наращивания выразительных возможностей языка SCn. При необходимости расширения SCn новыми видами sc.n-полей достаточно описать образец sc.n-поля и соответствующее правило трансляции этого поля в текст SCs без необходимости написания или изменения описания грамматики языка и построения новой версии транслятора sc.n-текстов.

*действительное рациональное число*

∈ Предметная область теории арифметики

⊃ действительное целое число

⊃ натуральное число

§ Отношения, области определения которых являются надмножеством описываемого множества и каждая связка которых связывает элементы описываемого множества между собой:

- *неравенство действительных рациональных чисел\**
- *строгое неравенство действительных рациональных чисел\**

§ Отношения, области определения которых являются надмножеством описываемого множества и связки которых, в общем случае, связывают элементы описываемого множества с другими объектами:

- *сумма действительных рациональных чисел\**
- *произведение действительных рациональных чисел\**
- *минимум действительных рациональных чисел\**
- *максимум действительных рациональных чисел\**

**Рис. 3. Пример sc.n-статьи.**

Средствами языка C++ и библиотеки CURL разработаны программные средства автоматической сборки полученных таким образом SCs фрагментов и их конвертации в базу знаний. Предложенная модель сочетает особенности

условно-типовой и формализованной модели гипертекста [1] и позволяет представлять знания и описывать онтологии в гипертекстовом виде.

#### **ЛИТЕРАТУРА**

1. Башмаков, А.И., Башмаков, И.А. Интеллектуальные информационные технологии: Учеб. пособие. — М., 2005.
2. Голенков, В.В. Представление и обработка знаний в графодинамических ассоциативных машинах. — Минск, 2001.
3. Колб, Д. Г. Модель интерфейса графодинамической ассоциативной машины / Известия Белорусской инженерной Академии. — Минск, 2003. № 1 (15) / 1. — С. 188 — 191.
4. Ивашенко В.П. Унифицированное представлени и интеграция знаний / Материалы Международной научн.-техн // Конференции OSTIS, 2013: Минск, Республика Беларусь, БГУИР 21-23 февраля 2013. — Минск, 2013.
5. Горбатов В.А. Фундаментальные основы дискретной математики. Информационная математика. — М., 2000.